CORRELATION of PROTOCOL CLOCKS

SRNTN 58

JULY 1988

Prepared by: Sue Cline
James A. Stevens
Edited by: Neil Gower
Collins Defense Communications
Rockwell International
Richardson, Texas 75081

Prepared for: Major Mark Pullen
Major Brian Boesch
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209-2308

Mr. Paul Sass
US Army CECOM
Attention: AMSEL-RD-C3-AC-B
Fort Monmouth, New Jersey 07703-5202

Sponsored by: Defense Advanced Research Projects Agency (DoD)
Information Sciences and Technologies Office
Survivable Adaptive Networks (SURAN) Program
ARPA Order No.6067/9142
Issued by NRCC under Contract # N00140-87-C-8903

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30 1986

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b RESTRICTIVE MARKINGS:

2a SECURITY CLASSIFICATION AUTHORITY: DD FORM 254 DATED JUNE 11, 1986

2b DECLASSIFICATION / DOWNGRADING SCHEDULE: CG-45

3 DISTRIBUTION / AVAILABILITY OF REPORT:

4 PERFORMING ORGANIZATION REPORT NUMBER(S):

5 MONITORING ORGANIZATION REPORT NUMBER(S): SRNTN 58

6a NAME OF PERFORMING ORGANIZATION: ROCKWELL INTERNATIONAL COLLINS DEFENSE COMMUNICATIONS

6b OFFICE SYMBOL (If applicable): M/S 460-340

7a NAME OF MONITORING ORGANIZATION: DEFENSE ADVANCED RESEARCH PROJECTS AGENCY INFORMATION SCIENCES & TECHNOLOGIES OFFICE

6c ADDRESS (City, State, and ZIP Code):
3200 E. RENNER ROAD
RICHARDSON, TEXAS 75081-6209

7b ADDRESS (City, State, and ZIP Code):
1400 WILSON BOULEVARD
ARLINGTON, VIRGINIA 22209-2308

8a NAME OF FUNDING / SPONSORING ORGANIZATION: DARPA

8b OFFICE SYMBOL (If applicable): ISTO

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: CONTRACT NO. N00140-87-C-8903

8c ADDRESS (City, State, and ZIP Code):
1400 WILSON BOULEVARD
ARLINGTON, VIRGINIA 22209-2308

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification): CORRELATION OF PROTOCOL CLOCKS

12 PERSONAL AUTHOR(S): CLINE, SUE AND STEVENS, JAMES A.

13a TYPE OF REPORT: SRNTN

13b TIME COVERED: FROM AUG 86 TO JUL 88

14 DATE OF REPORT (Year, Month, Day): 1988, JULY, 5

15 PAGE COUNT: 9

16 SUPPLEMENTARY NOTATION:

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
12	05.07	
25	02.05	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number):
TIME SYNCHRONIZATION, PACKET RADIO NETWORKS, CLOCK OFFSET, TIME CORRELATION, CLOCK SYNCHRONIZATION, EVENT SYNCHRONIZATION, CLOCK CORRELATION

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This SRNTN explores the ability and need to synchronize events in an LPR network. Event synchronization is discussed in terms of synchronizing time reference and/or correlating them. The use of different timestamping schemes is discussed. The synchronizing and/or correlating of events network-wide is necessary for both network control and network monitoring activities. Several methods of correlation and/or synchronization and their precision are reviewed and an implementation suggestion is made.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL:

22b TELEPHONE (Include Area Code):

22c OFFICE SYMBOL:

INTRODUCTION:

The ability to synchronize events in an LPR network is fundamental to the operation of the new protocols being designed. Synchronization of events may be accomplished by synchronizing time references or by correlating them. The following is an edited rendition of Jim Stevens' presentation to the SURAN implementors meeting held 3-5 September 1986.

DEFINITION OF TERMS:

SYNCHRONIZED	state in which clocks indicate the same time.
CORRELATED	state in which clocks indicate different times, but differ by a constant, known OFFSET.
RF CLOCK	clock used by IOP to transmit/receive code-slotted packets (Units = 5 microseconds).
8086 CLOCK	clock used by 8086 protocol and 8086 OS in the LPR (Units = 26.0416 microseconds).
RF TIME	time from RF CLOCK, synchronized within a subnet work.
UNIVERSAL RF TIME	time to which a network of several overlapping subnetworks, having different RF TIMEs, has been correlated.
8086 TIME	time from 8086 CLOCK used by protocol and OS within a single LPR.
UNIVERSAL 8086 TIME	8086 TIME which has been correlated by an OFFSET within a subnetwork or network (multiple subnetworks).

USE OF CORRELATED TIMESTAMP:

The LPR should use its 8086 CLOCK internally both for time tagging and for calculating time intervals.

When the LPR reports time tags externally it can report either:

1 - RF TIME
or
2 - UNIVERSAL 8086 TIME
or
3 - 8086 TIME and the RF TIME

Methods 1 and 2 require significant calculation and continual maintenance of the OFFSET by the LPR. Method 3 however shifts the burden of calculations and/or OFFSET maintenance to the external entity, e.g. the network monitor.

When an external entity puts a TIME into a packet intended for interpretation by an LPR it can put either :

1 - 8086 TIME
or
2 - UNIVERSAL 8086 TIME
or
3 - RF TIME

Method 1 requires that the external entity maintain a table of OFFSETs containing the OFFSET for each LPR in the network. This table would need to be continually updated to account for OFFSET changes in the LPRs as well as the entrance and exit of LPRs to and from the network. This method is highly impractical, requiring a high processing overhead and eliminating the use of broadcast packets.

Method 2 requires that each LPR maintain an OFFSET and perform its own correlation calculations. OFFSET maintenance would need to be performed at least every 8 minutes (explained in a later section).

Method 3 requires an LPR to calculate the difference in current RF TIME and externally specified RF TIMES and convert to local 8086 clock units for event scheduling. These conversions would be necessary only occasionally and do not involve an OFFSET calculation.

APPLICATIONS OF CORRELATED CLOCKS:

(1) Monitoring devices (such as the network monitor) could use RF TIME or UNIVERSAL 8086 TIME to correlate the Monitoring Data Packets (MDPs) or other information it receives from the LPRs out in the net.

(2) Monitoring devices (such as the network monitor) could use RF TIME or UNIVERSAL 8086 TIME to command an LPR to clear its cumstats at time X, take a snapshot at time Y, and report the data back to the device at time Z. This would allow a test to be run between time X and Y without any extraneous monitoring traffic being broadcast to skew the test data.

```
   |                                |     different      |
   |                                |    for each LPR    |
   |<--------- run test ----------->|<- to reduce net ->|
   |                                |     congestion     |    UNIVERSAL
---|--------------------------------|--------------------|------>  8086
   X                                Y                    Z         TIME

 LPRs clear                     LPRs take           LPRs report
 cumstats                       snapshot            snapshots
```

(3) Network managing devices (such as a network control center) could use RF TIME or UNIVERSAL 8086 TIME to command LPRs to change network-wide parameters at time T. For example, the network could be told to change frequency at time T and all of the LPRs would change over at the same time. Thus there would not be a substantial period of time when the PRNET would be partitioned as would be if a scheme were used that did not require correlated clocks.

HOW TO CORRELATE CLOCKS:

Following are the steps required to correlate the network clocks.

(1) Correlate or synchronize the RF CLOCKs:

If there is only one subnetwork, then synchronize RF CLOCKs as described in David Young's SRNTN 29.

RF TIME = RF CLOCK

If there is more than one subnetwork and the subnetworks have different RF CLOCKs, then correlate subnetwork time to a UNIVERSAL RF TIME and synchronize RF CLOCKs within each subnetwork to the subnetwork time.

```
UNIVERSAL RF TIME = RF TIME + RF_OFFSET
                  = RF CLOCK + RF_OFFSET
```

(2) Find the OFFSET between the 8086 CLOCK and the RF CLOCK:

If the LPR is maintaining the OFFSET, it must do the calculation periodically and should update the OFFSET every time it has to perform a coarse synchronization. The LPR could determine the OFFSET by requesting the IOP to read the RF CLOCK and then determining the OFFSET between the returned RF TIME and the current 8086 TIME. Note that a conversion from the RF TIME units to the 8086 TIME units is first required, since the RF CLOCK is kept in 5 us ticks and the 8086 CLOCK is kept in 26.0416 us ticks.

If the external entity is maintaining the OFFSET, then it would do the calculation every time it received a timestamp from an LPR consisting of 8086 TIME and RF TIME.

(3) If there is only one subnetwork:

```
UNIVERSAL 8086 TIME = RF TIME
                    = RF CLOCK
                    = 8086 CLOCK + OFFSET
```

If there are more than one subnetwork:

```
UNIVERSAL 8086 TIME = UNIVERSAL RF TIME
                    = RF TIME + RF_OFFSET
                    = 8086 CLOCK + OFFSET + RF_OFFSET
```

PRECISION OF CORRELATED CLOCKS:

The reference for the correlated time clocks is the RF CLOCK. The RF CLOCKs within a network can be synchronized (across a neighbor link) to within 5 microseconds using FINE synchronization. Note that the synchronization error across a network is naturally expected to be larger than across a single link; however, it should still be an order of magnitude less than a millisecond.

The error in correlating the UNIVERSAL 8086 TIME will come from the random delays introduced by performing an IOP call to read the RF CLOCK and correlating it with the 8086 TIME. While the correlation calculation, with different units of measure for the two clocks, will introduce some error, the primary error component will be the uncertainty associated with the RF TIME received back from the IOP compared to the 8086 TIME due to interprocessor call/return delays (see timeline below). The delay can be minimized by having the LPR OS timestamp the IOP call with 8086 TIME at the time of the call and at the time the IOP returns to the LPR OS for scheduling of the protocol job. This technique would eliminate delays due to the transfer of processing between the protocol and the LPR OS and the delays the LPR OS will encounter when the IOP is busy, since the first timestamp would be the time at which the IOP channel is available. Should the IOP delay the return from the call, the delayed_by_receive indication will be set. This sample should be discarded by the protocol and the IOP call issued again. Having both timestamps allows saving the lowest delay encountered/expected for a "good" IOP call (with no extra delays). The protocol would discard any reading which included a delay greater than (C-A)minimum + variance. Variance could be on the order of 20%. The filtered random delay is expected to be on the order of a few milliseconds and can be further compensated for by performing many calls and taking an average, but the gain in precision may not be worth the extra processing time overhead. However several calls may be desirable after initialization to establish the lowest expected delay value.

```
                                IOP task block runs:
                                 - Reads RF TIME;
                                 - Returns to LPR OS
                                             |
IOP --------------------------|------------------------------> RF TIME
                                             B
                           A                      C
LPR -|--------------|---------------------|-----------------|-> 8086 TIME
     |              |                     |                 |
LPR job runs:  LPR OS runs:        LPR OS runs:        LPR job runs:
-Queues time   -Calls IOP          -Stamps 8086 TIME   -Verifies sample
  request      -Stamps 8086 TIME   -Schedules LPR job  -Calculate OFFSET
```

Two simple ways to calculate OFFSET between RF CLOCK and 8086 CLOCK:

(1) Ignore the task block call time or return time and treat either time A or C as occurring at the same instant as B.

$$B - A = OFFSET \quad \text{or} \quad B - C = OFFSET$$

(2) Consider the task block call and return times to be about equal.

$$B - \frac{A + C}{2} = OFFSET$$

More complicated ways to calculate the OFFSET would require multiple calculations of the OFFSET and taking the average of the different OFFSETs.

The drift of the RF CLOCK is less than 1 part per million. The drift of the 8086 CLOCK is less than 1 part per ten thousand. Let's assume that we desire that the clocks be correlated to within 0.1 second, which implies that each LPR could be off by + or - 50ms:

* RF CLOCK

  At 1 part per million, the RF CLOCK could drift 50ms after 50,000 seconds

  50,000 secs = 13.89 hours

* 8086 CLOCK

  At 1 part per ten thousand, the 8086 CLOCK could drift 50ms second after 500 seconds

  500 secs = 8.33 mins

Thus, if the LPR is maintaining the OFFSET, it would have to correlate the 8086 CLOCK with the RF CLOCK about once every 8 minutes, assuming that the RF CLOCK would be correlated much more often than every 14 hours. It is desirable to maintain a much more accurate correlation than 50ms should transmission scheduling be implemented (scheduled transmissions are synchronized by the protocol using 8086 TIME). Should an accuracy on the order of the uncertainty of the RF TIME to 8086 TIME correlation be desired (a few miliseconds), then the correlation should occur about ten times as often or every minute (actually 0.83 min).

Suggested Implementation:

It is suggested that time correlation within the network be accomplished using existing features. The current thinking is that the LPR should be burdened with as little extraneous processing as possible. This goal may be reached by using the UNIVERSAL RF TIME.

RF TIME sychronization is already in place in the current code (Refer to SRNTN 29 "SURAP Network Time Synchronization"). It is accomplished using timestamps on the PROPs which occur every 7.5 seconds.

In an implementation scheme where the LPR is maintaining the OFFSET, the OFFSET calculation must be performed by every LPR in the network whenever a coarse RF TIME change is made (possibly every 7.5 seconds) AND every 8.33 minutes (to compensate for drift between the RF and 8086 CLOCKS). This extra processing burden may be reduced by sending the host the necessary data (8086 TIME and RF TIME) to calculate the OFFSETs if needed. It is suggested that the MDP (Monitoring Data Packet) header be modified to include an RF TIMESTAMP in addition to the 8086 TIMESTAMP which is already included. This single modification would present the host with all the necessary data to calculate the UNIVERSAL 8086 TIME.

This is not to imply that the individual LPR does not need the ability to correlate its individual 8086 and RF TIMEs. In the case of host-directed future events , the host should send times for future events in RF TIME and each LPR would calculate the necessary 8086 TIME and schedule the event using this calculated time.

```
8086-TIME-for-task = [8086-TIME-when-request-received  +
          ( (RF-TIME-for-task  -  RF-TIME-when-request-received) /
                    (26.0416 / 5) )]
```

In this way the LPR is not burdened with OFFSET calculations except when scheduling host-directed future events. This situation would be infrequent when compared to the frequency of LPR-to-host data transmissions.

SUMMARY:

This paper has addressed the subject of time correlation within the LPR network. After reviewing the pros and cons of maintaining a UNIVERSAL 8086 TIME, it is suggested that the RF TIME be used instead. Specifically:

(1) It is suggested that time correlation within the network be accomplished using existing features. The current thinking is that the LPR should be burdened with as little extraneous processing as possible.

(2) RF TIME sychronization is already in place in the current code (Refer to SRNTN 29 "SURAP Network Time Synchronization"). This virtually eliminates the need for the correlation of 8086 TIMEs within the network since all nodes are synchronized to one RF TIME .

(3) An RF TIMESTAMP should be included in MDP packets and correlation calculations of UNIVERSAL 8086 TIMES should be done by the requesting host, in most cases. In this way the LPR is not burdened with OFFSET calculations except when scheduling host-directed future events. This situation would be infrequent when compared to the frequency of data transmission from LPR to host.

(4) The suggested implementation scheme would require little extra software, thereby consuming less LPR memory and allowing for faster implementation.

(5) The current testing mechanism, (PC-NETSIM) does not require the correlation of 8086 and RF TIMEs and the current development status of the LPR radio would indicate that the actual correlation scheme may be unnecessary. Should the actual correlation prove to be necessary, however, it could easily be added to the current scheme at a later date.